На второй стадии – стадии анализа угроз и уязвимостей:

- оценивается зависимость пользовательских сервисов от определенных групп ресурсов;
- оценивается существующий уровень угроз и уязвимостей;
- вычисляются уровни рисков;
- анализируются результаты.

Группировка ресурсов производится с точки зрения угроз и уязвимостей.

Оценка уровней угроз и уязвимостей производится на основе исследования косвенных факторов. Программное обеспечение CRAMM для каждой группы ресурсов и каждого из 36 типов угроз генерирует список вопросов, допускающих однозначный ответ.

Уровень угроз оценивается, в зависимости от ответов, как очень высокий, высокий, средний, низкий и очень низкий.

Уровень уязвимости оценивается, в зависимости от ответов, как высокий, средний и низкий.

Возможно проведение коррекции результатов или использование других методов оценки.

На основе этой информации рассчитываются уровни рисков в дискретной шкале с градациями от 1 до 7.

Полученные уровни угроз, уязвимостей и рисков анализируются и согласовываются с заказчиком. Только после этого можно переходить к третьей стадии метода.

На третьей стадии – стадии выбора контрмер – CRAMM генерирует несколько вариантов мер противодействия, адекватных выявленным рискам и их уровням. Контрмеры можно объединить в три категории: около 300 рекомендаций общего плана; более 1000 конкретных рекомендаций; около 900 примеров того, как можно организовать защиту в данной ситуации.

На этой стадии можно провести сравнительный анализ эффективности различных вариантов защиты.

Достоинством этого метода является то, что можно достаточно быстро провести анализ риска и у результатов анализа будут достаточно четкое обоснование.

Недостатком этого метода является то, что на некоторых этапах могут использоваться нечеткая и субъективная информация. И, кроме того, этот метод может использоваться только для достаточно стандартных информационных систем, т. к. только для них можно выбрать вариант данного продукта с полностью подходящим перечнем критериев.

Сейчас появилась новая идея нового подхода – теоретико-графового подхода к анализу рисков. Подобный подход позволяет использовать математическую модель в качестве опорного инструмента для доказательства истинности того или иного утверждения, касающегося степени защищенности и инертности исследуемой системы защиты. Однако этот подход пока только теоретический и не имеет практической апробации.

Из всего вышесказанного можно сделать вывод, что ни одна из существующих методик анализа рисков и, тем более, программных продуктов на их основе не являются совершенными. У всех подходов есть свои недостатки, а у некоторых достаточно серьезные. Поэтому актуальным является развитие теории анализа рисков в направлениях:

- разработки математических методов моделирования компьютерных систем с целью анализа рисков;
- практической апробации существующих методов анализа рисков в условиях нечеткого субъективного представления информации о быстро меняющихся технологиях обработки информации;
- разработки программного продукта, оперативно реализующего методики анализа рисков, учитывающих изменяющиеся условия.

УДК 004.056.5

АНАЛИЗ ЗАЩИЩЕННОСТИ МОБИЛЬНЫХ ОПЕРАЦИОННЫХ СИСТЕМ

Денис Кудин, Владислав Корольков*
Центр информационной безопасности,
** Запорожский национальный технический университет*

Аннотация: **Анализируются особенности реализации мобильных операционных систем с точки зрения их безопасности. Определяются возможные угрозы, способы внедрения, распространения и исполнения вредоносного кода. Рассматриваются методы формального анализа поведения подсистемы защиты во времени и оценки ее реакции на угрозы вредоносного кода с учетом особенностей упрощенной реализации механизмов защиты в карманных персональных компьютерах.**

Summary: **Analyses peculiarities of mobile operating systems' realization from the point of view of their security. It determines possible threats, means of intrusion, distribution and execution of malicious code. It also proposes the methods of formal analysis of protection subsystem behavior and assessment of its reaction on malicious code threats taking into account the peculiarities of simplified security mechanisms in PDAs.**
Ключевые слова: **Мобильная операционная система, подсистема защиты, безопасность, КПК, PDA.**

I Введение

Под термином "мобильные операционные системы" будем подразумевать те операционные системы, которые предназначены для использования в карманных персональных компьютерах (КПК, PDA), например, Palm OS, EPOC/Symbian, Microsoft Windows CE/Pocket PC/Windows for handheld PC, Newton и другие.

В настоящее время КПК приобретают все большее распространение. Помимо использования в личных целях, они широко применяются в корпоративных сетях для решения различных производственных задач, имея ряд неоспоримых преимуществ по сравнению с традиционными настольными компьютерами. Однако PDA свойственны и весьма существенные недостатки, связанные, в первую очередь, со слабой реализацией их механизмов безопасности, о чем компании-производители, как правило, умалчивают.

Данная статья посвящена анализу защищенности мобильной операционной системы Palm OS, поскольку, по данным [1], портативные устройства на платформе Palm являются наиболее распространенными и занимают на сегодняшний день около 80% рынка карманных компьютеров. Кроме того, некоторые слабые стороны Palm являются характерными для всех типов платформ.

С появлением PDA возникли новые типы угроз, не свойственные настольным компьютерам. Портативные устройства, имеющие свою конструкцию электрических схем, специфические свойства, характерные для разных моделей PDA, встроенную операционную систему, средства беспроводной связи и использующиеся в корпоративной информационной инфраструктуре, все чаще становятся объектом промышленного шпионажа, хищений, атак и различных угроз вредоносного кода (вирусов, троянских коней и червей). Архитектура Palm-ориентированных устройств предрасполагает к этому.

II Общая архитектура Palm-устройств

На самом высоком уровне абстракции архитектуру Palm-устройств, а также PDA, работающих на других платформах, можно разделить на четыре уровня [2] – приложения, операционная система, API и драйвера, аппаратная часть, – (см. рис. 1).

Рисунок 1 – Типовая уровневая архитектура PDA

Использование Palm OS API (Application Programming Interface) предоставляет независимым разработчикам приложений возможность в определенной степени абстрагироваться от аппаратной части устройства. Если приложение корректно использует API, то для его переноса на устройство, работающее под управлением Palm OS на другой аппаратной конфигурации, потребуется лишь рекомпиляция этого приложения.

Palm OS позволяет приложению получить доступ к аппаратной части, минуя уровень программного интерфейса, т. е. приложение может напрямую обращаться к процессору и памяти. С одной стороны, это дает возможность разработчику более полно использовать возможности оборудования. Но, с другой стороны, во-первых, такая возможность отрицательно повлияет на переносимость приложений, а, во-вторых, прямой доступ к процессору открывает путь злоумышленникам к ресурсам устройства, т. е. путь "Б" является уязвимым. В идеальном случае, в целях контроля доступа и безопасности, только сама операционная система должна иметь доступ к низлежащему уровню аппаратной части, а все приложения должны работать через API.

Palm OS разрабатывалась как открытая и модульная система для поддержки приложений, разработанных третьими сторонами. В ней отсутствует такое понятие как контроль доступа на основе уровней или файлов. Код программ и все данные доступны для чтения и модификации любым пользователям или приложениям. При таком "универсальном" механизме доступа к памяти невозможно отличить вредоносную программу от легальной на основании только лишь операций чтения/записи в памяти и системных вызовов.

Среди прочих особенностей Palm OS следует отметить следующие [3]:

- Palm OS не использует традиционную файловую систему. Информация хранится во фрагментах памяти, называемых "записями", которые группируются в "базы данных". Такая база данных является аналогом файла. Данные разбиваются на множество записей, вместо того чтобы храниться в виде одной непрерывной цепочки;
- приложения в Palm OS состоят, как правило, из одного потока. Каждое приложение содержит функцию PilotMain, которая аналогична функции main в программах, написанных на языке С. Для загрузки приложения операционная система вызывает функцию PilotMain и передает ей код загрузки, определяющий, должна ли программа взаимодействовать с пользователем. Назначением функции PilotMain является получение кодов загрузки и ответ на них. Предполагается, что в дальнейшем будет возрастать количество многопотоковых приложений;
- приложения могут посылать коды загрузки друг другу. Следовательно, можно запускать одно приложение из другого для выполнения определенных действий и/или модификации тех или иных данных.

III Эффективность парольной защиты

Пользователь имеет возможность установить пароль с помощью приложения Security (Безопасность). В Palm OS версий ниже 4.0 существует опасность перехвата и дешифровки пароля пользователя, т. к. используется слабая схема защиты [2].

Максимальная длина пароля составляет 31 символ, причем, независимо от реальной длины пароля, результирующий кодированный блок будет иметь длину 32 байта. Используется два метода кодирования паролей. Если пароль состоит из четырех или менее символов, то вычисляется индекс на основании длины пароля и выполняется операция XOR ("исключающее ИЛИ") над строкой и фиксированным 32-байтовым блоком, показанным на рис. 2.

```
09 02 13 45 07 04 13 44 0C 08 13 5A 32 15 13 5D
D2 17 EA D3 B5 DF 55 63 22 E9 A1 4A 99 4B 0F 88
```

Рисунок 2 – Фиксированный 32-байтовый блок для кодирования паролей длиной 4 и менее символов

Если пароль имеет длину более четырех символов, то строка расширяется до 32 байт и проходит через четыре цикла операции XOR с фиксированным 64-байтовым блоком, показанным на рис. 3.

```
B1 56 35 1A 9C 98 80 84 37 A7 3D 61 7F 2E E8 76
2A F2 A5 84 07 C7 EC 27 6F 7D 04 CD 52 1E CD 5B
B3 29 76 66 D9 5E 4B CA 63 72 6F D2 FD 25 E6 7B
C5 66 B3 D3 45 9A AF DA 29 86 22 6E B8 03 62 BC
```

Рисунок 3 – Фиксированный 64-байтовый блок для кодирования паролей длиной более 4-х символов

В процессе обмена данными между настольным компьютером и Palm устройством с помощью HotSync используется протокол SLP (Serial Link Protocol). Один из SLP пакетов содержит структуру, в которой находится кодированный блок пароля. Подробности и алгоритмы декодирования паролей известны и описываются в [2].

Рекомендации по повышению эффективности защиты:

- использовать Palm OS версии не ниже 4.0.;
- ввести механизм запроса/ответа (challenge/response), что уменьшит вероятность перехвата противником структуры с паролем при использовании пассивного мониторинга транспортной среды;
- при вычислении хэша паролей использовать такую информацию, как имя пользователя Palm, идентификатор пользователя, уникальный серийный номер устройства, что позволит исключить вероятность представления одного и того же пароля с одинаковым хэш-кодом на разных устройствах;
- использовать политику блокирования и шифрования данных. Приложение Security поддерживает

возможность блокирования системы, т. е. устройство не будет работать, пока пользователь не введет верный пароль. Шифрование данных может быть осуществлено с использованием большого количества приложений разработки третьих сторон;

- внедрить альтернативную схему парольной защиты. Существуют независимые решения, предоставляющие возможность парольной защиты устройства при включении, требующей ввода рукописной подписи, графических паролей, установки аппаратного ключа и др.

IV "Черные ходы" для отладки приложений

В Palm OS встроено средство, называемое "Palm Debugger" и предназначенное для отладки приложений и администрирования баз данных, находящихся на физическом устройстве.

При нажатии определенной короткой комбинации клавиш устройство входит в один из двух интерфейсов отладчика и отслеживает обмен данными через последовательный порт по интерфейсу RS-232. "Консольный режим" взаимодействует с отладчиком на высоком уровне и используется в основном для манипуляций с базами данных. "Режим отладки" используется для ассемблерного и регистрового уровня отладки. Горячая перезагрузка устройства выведет его из режима отладки, не оставляя при этом никаких следов работы последнего.

При использовании Palm OS версий ниже 4.0 отладчик может быть активирован даже при включенной блокировке устройства. Это дает злоумышленнику возможность извлечения блока с кодированным системным паролем, а также получения доступа к информации о всех базах данных и записях, хранимых в устройстве.

Поскольку режимы отладки работают через последовательный порт, возможно создание приложения на основе Palm OS для эмуляции команд отладчика и, с помощью модифицированного HotSync кабеля, использование этого приложения для получения паролей и другой важной информации с PDA.

V Механизмы внедрения, распространения и исполнения вредоносного кода

Любой метод загрузки информации в Palm устройство может послужить точкой входа для проникновения вредоносного кода. Существует четыре основные точки входа:

- операция синхронизации HotSync;
- последовательный порт;
- инфракрасный порт;
- беспроводная радиосвязь.

Кроме того, программы, в том числе вредоносные, могут быть загружены через Palm Debugger, о чем упоминалось выше.

Процедура инсталляции дополнительных приложений является довольно простой. В комплекте ПО Palm Desktop поставляется утилита Install, которая копирует требуемое приложение в каталог /Palm/<user>/Install настольного компьютера. При следующей процедуре HotSync содержимое этого каталога автоматически переносится на PDA. Это может служить одним из примеров кросс-архитектурного вирусного распространения. В течение операции HotSync не предусмотрены механизмы подтверждения и аутентификации. Следовательно, вопрос безопасности настольного компьютера, как неотъемлемого звена цепочки обмена информацией с КПК, является ключевым. Если настольный компьютер скомпрометирован, то КПК также может считаться скомпрометированным.

Для защиты от описанного механизма внедрения вредоносного кода наиболее эффективным является тщательная визуальная проверка содержимого каталога /Palm/<user>/Install перед операцией синхронизации. Существенно уменьшить риск заражения позволит криптографическая цифровая подпись приложений производителями и проверка этой подписи пользователем перед установкой на свой КПК.

Каналы ("conduits"), в форме динамических DLL библиотек, взаимодействуют с программой HotSync Manager настольного компьютера. Они осуществляют передачу данных между Palm OS устройством и определенным настольным приложением в течение операции синхронизации HotSync. Стандартные каналы Palm OS передают данные приложений Address, Date Book, Memo Pad и To Do List программному обеспечению Palm Desktop. Данные приложения Palm Expense напрямую взаимодействуют с Microsoft Excel. Существуют также каналы разработки третьих сторон, которые подменяют стандартные каналы и обеспечивают интерфейс с программами Microsoft Outlook или Exchange, Lotus Notes, Novell GroupWise или другими PIM-приложениями (Personal Information Manager).

Каналы являются чрезвычайно удобным способом для распространения вредоносного кода. Помимо вирусного заражения (например, макро-вирусом через Microsoft Word или Excel), вредоносный код, переданный с Palm устройства на хост-компьютер через канал, может реализовать известную уязвимость

настольного приложения, что скомпрометирует компьютер (например, вызовет исполнение произвольного кода, утечку информации, отказ в обслуживании, повышение привилегий и т. д.).

Каждое приложение, работающее под управлением Palm OS, имеет свой 4-байтовый **код создателя CID** ("creator ID"), используемый в целях идентификации. CID встроенных приложений имеют вид: Address – addr, Calculator – calc, Date Book – date, Expense – exps, HotSync – sync, Mail – mail, Memo Pad – memo, Preferences – pref, Security – secr, To Do List – todo. Если CID вредоносной программы будет совпадать с одним из встроенных идентификаторов, то такая программа будет выполнена вместо встроенного приложения. Например, если присвоить троянской программе "A" идентификатор создателя calc, то каждый раз при вызове пользователем встроенной программы калькулятора вместо нее будет выполняться "A".

Подобный механизм поведения характеризуется типом буфера LIFO (Last In First Out). При инсталляции нового ПО в систему, его CID добавляется в хвост буфера. Перед запуском программы буфер сканируется в обратном порядке, начиная с хвоста, на предмет поиска соответствующего идентификатора CID. Когда находится первое совпадение, поиск прекращается.

Производители ПО могут решить проблему подмены CID на уровне операционной системы путем отслеживания и запрета дублирующихся идентификаторов создателя. Более того, необходимо производить поиск по всему списку CID и если будут найдены дублирующиеся идентификаторы, ни одно из соответствующих приложений не должно быть запущено без санкции пользователя.

Для соединений типа точка-точка на очень близкие расстояния обычно используется **связь по инфракрасному (IR) порту**. В стандартном сеансе связи по IR порту Palm OS посылает код загрузки sysAppLaunchCmdExgAskUser принимающему приложению. Обычно приложения не имеют собственных обработчиков, для этого кода загрузки и в данном случае срабатывает стандартный механизм, который выдает пользователю диалоговое окно с запросом на подтверждение или отмену сеанса связи. Если же приложение поддерживает обработку данного кода загрузки и установит результирующий флаг в состояние exgAskOk, то приложение отошлет код загрузки sysAppLaunchCmdExgReceiveData и будет всегда принимать любые входящие данные без запроса подтверждения пользователя. Используя подобным образом возможности программы Exchange Manager, канал инфракрасной связи становится удобной средой для внедрения вредоносного кода с персонального компьютера на КПК и наоборот.

Как уже было сказано, инфракрасная связь применима на малых расстояниях для обмена информацией между двумя находящимися рядом устройствами. Если же необходима связь на большие расстояния и использования услуг Интернет, то применяется **беспроводная радиосвязь**, которая также может служить средой передачи вирусов, троянских программ и других типов вредоносного кода.

Возможные методы хранения исполнимого кода вредоносной программы в Palm OS:

- базы данных и настройки. Программный интерфейс Palm OS имеет диспетчер параметров и данных (Preferences and Data Manager), предоставляющий ряд функций для доступа к настройкам системы и приложений, а также для манипуляции с базами данных. Эти функции, ввиду отсутствия разграничения прав доступа к записям баз данных, могут применяться для сохранения вредоносного содержимого, например, в полях записей, которые пусты и не используются приложениями;

- флэш-память. На флэш-памяти можно хранить данные за пределами адресного пространства системы, но в пределах действительной карты памяти, заданной регистрами Group-Base Address (для архитектуры DragonBall). Стандартные приложения, использующие функции API, не смогут обратиться к информации, расположенной в такой области. Подобным образом можно скрывать вредоносный код от антивирусных программ.

Возможные методы загрузки вредоносного кода на исполнение:

- немедленное явное исполнение сразу же после проникновения в систему;

- отложенное исполнение с "инкубационным" периодом. Вредоносный код может исполняться по наступлению ожидаемого критерия – определенной даты, нажатия комбинации клавиш, поступления системного сообщения и т. д. Это существенно затрудняет определение времени и обстоятельств заражения. Наиболее опасным является использование в качестве инициирующего критерия определенных кодов загрузки, отсылаемых системой для всех приложений, поскольку программный код, расположенный в сегментах, на которые указывают соответствующие обработчики кодов загрузки, выполняется прозрачно для пользователя;

- неявное исполнение с перехватом стандартных системных вызовов. Атакующий может подменить вектор прерывания для определенной системной функции таким образом, что он будет указывать на вредоносный код.

VI Некоторые подходы к формальному анализу системы защиты

Можно считать, что подсистема защиты КПК является динамической системой с конечным множеством сосредоточенных параметров. Ее поведение во времени описывается системой обыкновенных дифференциальных уравнений (задача Коши), которая, в общем случае, может быть представлена так:

$$F(t, x, p^{(1)}, p^{(2)}, \text{K}\ , p^{(r)}) = 0\ , \tag{1}$$

где F – система функций, описывающих текущую конфигурацию подсистемы защиты,
x – вектор переменных состояния,
$p^{(i)}$ – вектор производных по t от x порядка i .

К этим уравнениям необходимо добавить еще систему начальных условий.

Довольно часто систему уравнений (1) удается разрешить относительно производных. Тогда эта задача принимает следующий вид:

$$\begin{aligned} \frac{\partial x}{\partial t} &= F(t,x)\ , \\ x(0) &= x^{(0)}\ . \end{aligned} \tag{2}$$

Рассмотрим два метода решения задачи Коши (2) – метод Эйлера с пересчетом и многошаговый метод прогноза и коррекции.

Воспользуемся разложением искомой функции $x(t)$ в степенной ряд в окрестности точки x_k , сохранив при этом 3 первых члена разложения:

$$x_{k+1} = x_k + h\,x'_k + \frac{h^2}{2!}x''_k \tag{3}$$

Вторую производную запишем, воспользовавшись правой конечной разностью для $F(x,t)$:

$$x''_k = \frac{F(x_{k+1}, t_{k+1}) - F(x_k, t_k)}{h}\ .$$

Тогда из (3) получаем:

$$x_{k+1} = x_k + \frac{h}{2}\left[F(x_k, t_k) + F(x_{k+1}, t_{k+1})\right]. \tag{4}$$

Последняя формула используется итеративно: первый раз в правую часть в качестве x_{k+1} подставляют значение, вычисленное обычным методом Эйлера, затем используют вновь полученные значения.

Достоинства: самостоятельность метода; простота изменения шага интегрирования.

Недостатки: большой объем вычислений, выполняемых для достижения необходимой точности; оценку погрешности нельзя получить как попутный результат; требуются дополнительные (иногда значительные) вычисления.

Метод прогноза и коррекции является простейшим многошаговым методом решения задачи (2). Схема метода выглядит следующим образом:

прогноз: $$x^{(0)}_{k+1} = x_{k-1} + 2h\,F(x_k, t_k)\ ; \tag{5}$$

коррекция: $$x^{(i)}_{k+1} = x_k + \frac{h}{2}\left[F(x_k, t_k) + F(x^{(i-1)}_{k+1}, t_k)\right]\ ; \tag{6}$$

поправка: $$x_{k+1} = x^{(m)}_{k+1} + \frac{1}{5}\left(x^{(0)}_{k+1} - x^{(m)}_{k+1}\right)\ . \tag{7}$$

Этап "коррекция" выполняется итеративно m раз. Оптимальное число повторений коррекции $m=2$. Погрешность ограничения k-того шага получается как попутный результат:

$$\varepsilon_k = \frac{1}{5}\left(x^{(0)}_k - x^{(m)}_k\right). \tag{8}$$

Достоинства: метод экономичен по объему вычислений; погрешность интегрирования получается как побочный результат.

Недостатки: начинать решение задачи необходимо одношаговым методом; при изменении шага интегрирования в процессе решения задачи требуется временно возвращаться к одношаговому методу.

VII Выводы

Проанализированы особенности реализации мобильных операционных систем на примере наиболее распространенной системы Palm OS с точки зрения их безопасности. Определены возможные угрозы, способы внедрения, распространения и исполнения вредоносного кода, а также приведены рекомендации по защите от этих угроз.

Рассмотренные методы моделирования динамических систем и соответствующие выражения (4–8) могут применяться для формального анализа поведения подсистемы защиты во времени и оценки ее реакции на угрозы вредоносного кода с учетом особенностей упрощенной реализации механизмов безопасности в карманных персональных компьютерах.

Литература: 1. IDC, "Market Mayhem: The Smart Handheld Devices Market Forecast and Analysis, 1999–2004", Report 22430, June 2000. 2. Kingpin and Mudge. Security Analysis of the Palm Operating System and its Weaknesses Against Malicious Code Threats. @stake, Inc. 3. Palm, Inc., Palm OS Programmer's Companion, DN 3004–003. 4. McAfee.com, "Increased Protection for Wireless Users in Wake of Recent PDA Trojan Discovery", Press Release, September 5, 2001.

УДК 654.924

АЛГОРИТМ ФУНКЦИОНИРОВАНИЯ ТЕХНИЧЕСКИХ СРЕДСТВ СИСТЕМЫ БЕЗОПАСНОСТИ

Владимир Волхонский

Санкт-Петербургский Государственный университет аэрокосмического приборостроения

Аннотация: **На основе анализа обобщенной структурной схемы системы безопасности и ее основных элементов рассматривается последовательность принятия решений об обнаружении угроз составными частями системы. Предлагается математическая модель системы.**

Summary: **Based on general security system block diagram and its main element analysis of the sequence of decision about treats are accomplishing. Mathematical model of security system is offered.**

Ключевые слова: **Система безопасности, обнаружение угроз, модель, решение.**

Алгоритм функционирования системы охранно-пожарной сигнализации (ОПС), рассмотренный в [1], определяет логику работы ее основы – контрольной панели (КП) – в зависимости от состояния ее элементов и текущего режима работы. В то же время в ряде случаев необходимо выполнить анализ в более общем случае системы безопасности (СБ) и, как частный случай, системы ОПС с учетом анализируемых характеристик и параметров объекта для обнаружения угроз, алгоритмов их преобразования и принятия решений.

Развитие технологии и совершенствование алгоритмов обработки, повышение информативности элементов систем, с одной стороны, и усложнение функций систем безопасности, с другой стороны, привело к развитию такого нового направления, как многоуровневые системы принятия решений. С этой точки зрения можно выделить два основных типа СБ. Одноуровневые, когда решение об обнаружении требуемого события или угроз (нападение, возгорание и др.) и реакции системы на него принимается на одном уровне системы (в одном устройстве). Пример – автономный пожарный извещатель. После принятия решения о возгорании следует акустический сигнал тревоги. Многоуровневые системы, в которых решения об обнаружении угроз принимаются на различных уровнях системы. Например, срабатывание извещателя в системе охранной сигнализации (первичное решение о тревоге) не означает возникновения состояния тревоги. Это будет зависеть от ряда факторов (режим охраны или нет, используется ли алгоритм двойного срабатывания и т. п.), что определяется алгоритмом работы КП. Кроме того, даже при формировании сигнала тревоги панелью, окончательное решение может принимать центральная станция мониторинга.

Таким образом, с точки зрения принятия решений в составе некоторой системы охраны (централизованной, автономной, интегрированной и т. п.) имеют несколько уровней принятия решений о том или ином событии. Подобные системы, в которых как решения о регистрации тех или иных событий, так и решения о реакции элементов системы на эти события принимаются на различных уровнях будем называть системами с распределенными уровнями принятия решений или с распределенным «интеллектом».